



Morreale, Fabio and Moro, Giulio and Chamberlain, Alan and Benford, Steve and McPherson, Andrew P. (2017) Building a maker community around an open hardware platform. In: ACM Conference on Human Factors in Computing Systems (CHI 17), 6-11 May 2017, Denver, Colorado, USA.

**Access from the University of Nottingham repository:**

<http://eprints.nottingham.ac.uk/41912/1/Chamberlain%20music%20HCI.pdf>

**Copyright and reuse:**

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

This article is made available under the University of Nottingham End User licence and may be reused according to the conditions of the licence. For more details see:  
[http://eprints.nottingham.ac.uk/end\\_user\\_agreement.pdf](http://eprints.nottingham.ac.uk/end_user_agreement.pdf)

**A note on versions:**

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact [eprints@nottingham.ac.uk](mailto:eprints@nottingham.ac.uk)

# Building a Maker Community Around an Open Hardware Platform

Fabio Morreale<sup>1</sup>, Giulio Moro<sup>1</sup>, Alan Chamberlain<sup>2</sup>, Steve Benford<sup>2</sup> and Andrew P. McPherson<sup>1</sup>

Centre for Digital Music<sup>1</sup>  
School of EECS

Queen Mary University of London, UK  
[f.morreale, g.moro, a.mcpherson]@qmul.ac.uk

Mixed Reality Laboratory<sup>2</sup>  
School of Computer Science

The University of Nottingham, UK  
[alan.chamberlain, steve.benford]@nottingham.ac.uk

## ABSTRACT

This paper reflects on the dynamics and practices of building a maker community around a new hardware platform. We examine the factors promoting the successful uptake of a maker platform from two perspectives: first, we investigate the technical and user experience considerations that users identify as the most important. Second, we explore the specific activities that help attract a community and encourage sustained participation. We present an inductive approach based on the case study of Bela, an embedded platform for creating interactive audio systems. The technical design and community building processes are detailed, culminating in a successful crowdfunding campaign. To further understand the community dynamics, the paper also presents an intensive three-day workshop with eight digital musical instrument designers. From observations and interviews, we reflect on the relationship between the platform and the community and offer suggestions for HCI researchers and practitioners interested in establishing their own maker communities.

## ACM Classification Keywords

H.5.m. Information Interfaces and Presentation: Miscellaneous; H.5.1. Multimedia Information Systems: Evaluation/Methodology; H.5.5. Sound and Music Computing: Systems

## Author Keywords

Maker community, DIY, crowdfunding, embedded hardware, digital musical instruments, pluggable communities.

## INTRODUCTION

In recent years, HCI researchers have been studying the culture and activities of hacker and maker communities [3, 10, 24, 27, 26, 43, 44, 46]. Some such communities are physically collocated, for example at the hackspaces found in nearly every major city [27, 44, 46, 47], while other are online groups based

on shared interests or shared usage of particular technical tools [8, 22].

Most studies of maker communities have focused on established, stable communities [8, 24, 26, 28, 44, 46]. Comparatively little is known about how new maker communities emerge, partly because the identification of a group of people as a “community” can only be done with any certainty in retrospect. In particular, the process by which user groups develop around new digital tools, and how design decisions influence the dynamics of the user community, merit further study.

In this paper, we present a longitudinal study of the emergence of a community of makers using Bela [29], an open-source embedded hardware platform for creating musical instruments and interactive audio systems. We trace the motivation for the development of Bela, including its origins in a specific musical instrument project [49, 31], its expansion into a platform for teaching, its emergence as a prototyping tool for the digital musical instrument (DMI) community, and its dissemination through a successful crowdfunding campaign.

To better understand the dynamics of the emerging community of makers using Bela, we organised a 3-day workshop at STEIM, a prominent electronic music centre in Amsterdam. The workshop was organised specifically as a research tool

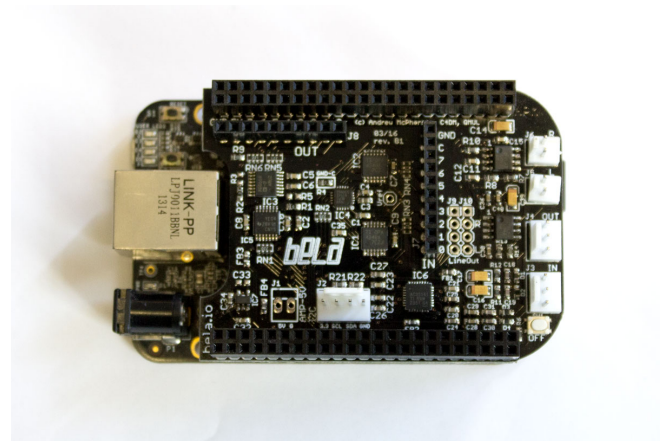


Figure 1. Bela, which consists of a custom hardware board (“cape”) on a BeagleBone Black running specialised software.

to understand the dynamics of the emerging Bela community [38]. Eight instrument designers ported their instruments to Bela, receiving colocated support from the developers. Regular observations and interviews with the participants were conducted to identify the reasons for attending and using Bela and the activities and interactions amongst participants.

From the development, crowdfunding, and workshop, we distil a set of reflections for HCI researchers who are interested in building communities around new tools. We examine the technical and user experience factors which users identify as important reasons for joining a community centred on a particular platform. We also reflect on the specific promotion, dissemination and community-building activities that are likely to generate a broad and sustainable community of users.

Though Bela is an open-source hardware platform, our results suggest that open-source release is not itself sufficient for a community to form. Instead, we find that the ability to connect Bela to other, established tools with their own communities is a powerful motivator for people to work with Bela and share their results. We suggest that Bela represents a case study in what we term *pluggable communities*: distributed communities of makers bound together by their use of a particular tool, where the communities grow in discrete leaps by connecting to other familiar tools, thereby leveraging their communities.

## BACKGROUND

The “maker community” is a term used to describe a community built on a collection of emerging technologies and practices [26], which are used to modify and repurpose existing materials to produce something new [10]. HCI researchers have been profoundly interested in understanding the maker movement from different perspectives: as technology-based democratising force that empowers consumers [1, 43, 26], as a venue where people collaboratively work on visions of the future through interaction design [27], and as an unexplored design area that should borrow tools and infrastructures that are typical of HCI [10, 24, 26, 33].

### Building Communities...

Maker culture is rooted in the hacker culture of the 60s-70s [25] and DIY amateur radio from the early 20th century [23], movements committed to peer production and to designing technologies that were open and user-modifiable. Makers inherit this alternative model of technology production, which is enabled by crowdfunding infrastructures and startups [27].

Crowdfunding actions can facilitate *community formation*. Product creators depend on their communities to secure funding and increase the awareness of their work, and they rely on community feedback to refine their products [14, 15]. Campaign funders (backers), beyond being interested in using the products, are keen to connect with a community, in helping others, and in support causes they trust [14, 15].

Once established, *community sustainability* mostly depends on the spontaneous involvement of its members to persist when the initial input from the developer reduces or vanishes. The motivations for people to voluntarily engage in a community

has been discussed in several research fields [21, 45]. Particularly relevant to this paper is the case of Open Source/Free Software (OS/FS) communities [2, 17]. People contribute to these communities for social and political reasons - defying economies of scale, allowing product viability without the necessity of mass production, rejecting proprietary software and private property rights [2] - as well as to develop individual skills and share knowledge [17]. In the arts domain, several OS/FS communities (e.g. Csound<sup>1</sup>, SuperCollider<sup>2</sup>) have reached a point where volunteer contributions complement or replace the actions of the original developers.

In maker communities, participants’ contribution is stimulated by the critical role that each member has, given the particularly varied backgrounds and skills of the members. Such a variety contributes to transfer ideas across communities, resulting in a rapid interdisciplinary skill building [23]. Among other motivations, makers participate as an opportunity for self-actualization [41], to get inspiration for future projects, to connect with people in the community and receive feedback on personal projects, and to educate others [23].

Of particular importance for the sustainability of some maker communities is a physical presence - *makerspaces*, which have been recently included in the agendas of HCI researchers as a means to build new research initiatives by working together with makers [27, 44]. These are spaces conceived for makers to physically gather to share and learn in a supporting environment [27]. Learning in particular is facilitated by the awareness of colleagues’ expertise and competence which are co-situated in a particular setting [42].

To this respect, Trainer et al. [47] analyse the benefits of situated co-worker familiarity (a “multiplex understanding that coworkers have of their counterparts in relation to themselves and their work together” [18]) in hackathons. Collocation is important for technical work as an opportunity to learn about each other’s interests and to watch experienced programmers code to understand their programming conventions [47].

Including expert facilitation in these spaces is crucial to establish creative goals and guide ideation, coordinate activities, and provide feedback [11]. The role of the facilitator is also devoted to “highlight key high-level characteristics or schemas exemplified by the idea, and provoke reconsideration of implicit assumptions about the design problem” [11].

### ...Around Open Hardware Platforms

Some maker communities are built around specific open-source hardware or software platforms. A noteworthy example is that of Arduino<sup>3</sup>. Besides the typical reasons that motivate people to contribute to OS/FS, the sustainability of this community is enhanced by the ability to add expansion boards, resulting in a great variety of extensions and variations [8].

One community whose formation has been documented is LilyPad, an Arduino variant designed to enable people to create their own e-textiles [7, 8]. Buechley and Hill [8] explain the

<sup>1</sup><http://www.csounds.com>

<sup>2</sup><http://supercollider.github.io>

<sup>3</sup><http://arduino.cc>

success of the platform through how it enables makers to build artefacts that were difficult to build before its introduction. They also focus on the distribution, adoption, and evolution of a community around the platform. Their investigation concluded that the LilyPad community consists of people who build artefacts with the kit, who document projects, develop and share tutorials, who develop modified LilyPad boards, and who contribute to the online forum. These activities in turn influenced the evolution of LilyPad itself.

### Digital Musical Instrument Communities

The platform described in this paper is situated within the field of DMI design, whose academic hub is the annual New Interfaces for Musical Expression (NIME) conference [19]. Marquez-Borbon and Stapleton [28] examine NIME from the perspective of communities of practice (CoPs) [48]. They conclude that, while NIME as a whole is heterogeneous and lacks some features of a CoP, several smaller CoPs exist within the NIME field. Amongst the four CoPs they identified is a community using Satellite CCRMA [4], an embedded platform for creating musical instruments that shares some features with Bela. Factors they identify as sustaining the Satellite CCRMA community include a repository with detailed documentation and an online forum that acts as a focal point for the exchange of ideas.

Community building comes up in a different context amongst DMI designers, whether or not they are part of the academic NIME community. As Jordà observed over a decade ago [20], many new instruments are invented each year, but very few of them attract a significant body of performers or repertoire. The process of building a musical community around a new instrument has been chronicled for the magnetic resonator piano [32] and the instruments forming the McGill Digital Orchestra [13]. Communities of composers and performers can exhibit certain similar dynamics to maker communities, including sharing of ideas and building on one another's results.

### Open Questions

This section discussed how topics related to maker community has been widely discussed in recent years in the HCI community. However, what specific design solutions (in terms of technological implementation and user experience) can stimulate community building around a tool have received comparatively little attention. Also, further study is needed on what specific activities, from initial ideation to product development to dissemination, are ideal for building a community.

### BELA: AN EMBEDDED INTERACTIVE AUDIO PLATFORM

This section gives an overview of Bela, an embedded platform for ultra-low-latency audio and sensor processing. Full technical details can be found in other papers [29, 35] and online;<sup>4</sup> here we describe the motivations and process of its development (Figure 2) in relation to community building.

### Initial Development

The platform that later became known as Bela was first created in early 2014 for the D-Box [49], a self-contained digital

musical instrument designed to be modified and hacked by the performer using circuit bending techniques [16]. The musical character of the D-Box is dependent on low-latency feedback loops between hardware and software [30], which set up the following technical requirements:

1. Embedded hardware device with smartphone-calibre CPU power (for complex real-time audio synthesis);
2. Stereo audio plus 8 high-resolution analog inputs and outputs (to create feedback loops);
3. Extremely low audio latency of 1 millisecond or less (compared to 5-20ms on typical computers);
4. Precise jitter-free alignment of audio and analog data.

Based on these requirements, the BeagleBone Black<sup>5</sup> single-board computer was chosen, and a custom hardware *cape* (expansion board), was designed. A software framework was developed based on Xenomai Linux<sup>6</sup> which simultaneously provided hard-real-time performance and access to high-level OS features. These design aspects are described in [29] and later became important in Bela's use by the community.

### From Instrument to Platform

The transition from the D-Box to Bela as a community platform came in two phases. The first phase began in late 2014 (highlighted red in Figure 2), in preparation for teaching a class on embedded digital signal processing at Queen Mary University of London (QMUL). The D-Box hardware cape was maintained, but the software was modified to expose a single function where the developer could write arbitrary code.

The needs of the class influenced the development of Bela and its eventual community uptake. Class needs included:

1. Low cost (since each student needed a kit);
2. Sufficient CPU power to run complex DSP algorithms in real time;
3. Simple, straightforward API for programming (minimum distraction in teaching);
4. Quick to get started, with minimum time spent configuring compiler toolchains and development environments.

Getting from the D-Box to a suitable teaching platform required particular attention to points 3 and 4. For the class, a compiler toolchain was pre-installed on the laboratory computers, but this requirement would later become an obstacle to community uptake.

### Towards a Bela Community

The second phase of the instrument-to-platform transition began in April 2015, when the hardware and software were released open-source and continued with a series of tutorials, workshops, and hackathons in London (highlighted green in Figure 2) held at QMUL and at established maker community venues such as London Music Hackspace. They ran 2-4 hours, were free to attend, and had the option to buy hardware at the end. Longer hack day events were also organised in which

<sup>4</sup><http://bela.io>

<sup>5</sup><http://beagleboard.org>

<sup>6</sup><http://xenomai.org>



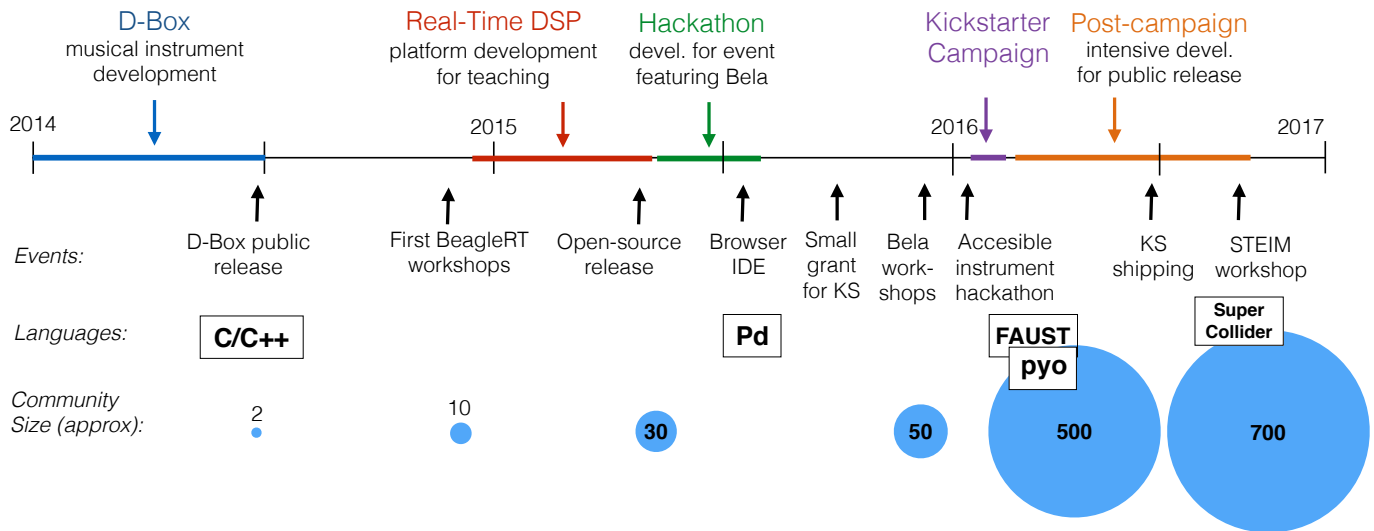


Figure 2. Timeline of Bela development, release and community activities.

participants used Bela to create new instruments. Together, these events established Bela within the local community, resulting in 20-30 regular users outside the university who provided us with feedback and ideas for future development.

Also starting April 2015, some students in the QMUL DSP class went on to use Bela in their own art and research. A few students contributed new features relevant to their own work, including an IDE (integrated development environment) which is served from the board but runs in the user's browser. The code could be compiled on the board, eliminating the need to install any specialised tools and significantly reducing the time to get started. Another contributed feature was support for the open-source graphical music programming language Pd<sup>7</sup>, which has a large and active user community (see *Porting* later in this paper). Many musicians and sound artists prefer to work in Pd than the lower-level C or C++.

### CROWDFUNDING CAMPAIGN

As the technical capabilities of Bela expanded alongside its user base, we set the goal of establishing a sustainable world-wide community using Bela for their own projects. The desired characteristics of this community included: a diverse mix of artists, engineers and makers; geographically decentralised but connected via a common set of online resources; self-supporting (at least in part), with community members contributing examples, support, and changes to the core code.

The motivation for creating a community came in part from our experience in the DMI field, where relatively few instruments are used beyond the first few performances [20, 32]. It also reflected our view that active communities are a cause, rather than effect, of successful maker tools.

Since Bela is a hardware platform, manufacturing in sufficient quantity for widespread distribution requires substantial up-front resources. Crowdfunding was a natural solution to this

bottleneck, and has the additional benefit of being perhaps the most common way to generate attention for new products aimed at the maker community [34]. Below we briefly describe the campaign structure and results in relation to the community building process.

### Structure

A Kickstarter campaign was launched in February 2016 to support the manufacturing of the Bela hardware. The campaign video and text focused on four core themes:

1. *Fast*: highlighting Bela's 1ms latency, which is significantly lower than other platforms.
2. *Connected*: highlighting the audio, analog and digital I/Os, plus the connectivity of the underlying BeagleBone Black (ethernet, USB, etc.).
3. *Embedded*: highlighting the self-contained, standalone nature of the board.
4. *Easy to get started*: featuring the built-in IDE, support for Pd and other user experience features.

The campaign page included both a high-level overview of Bela and its usability, aimed at a general readership, and detailed technical specifications for experienced developers. Campaign rewards included Bela capes, kits which included the BeagleBone Black and other parts to get started, and a limited number of D-Box instrument kits.

### Activities and Results

Prior to the campaign launch, we opened a mailing list for interested people to be notified of its start. List membership was via open voluntary subscription, and included previous attendees of our workshops, colleagues in the DSP and DMI communities, and others reached via professional networks and social media. When the campaign launched on 29 February 2016, it was announced to 391 subscribers on the list.

<sup>7</sup><http://puredata.info>

The campaign launch was also promoted through social media (Facebook and Twitter), contacts to online and print media, and individual email contacts. No paid advertising was conducted. Unlike some campaigns which attract substantial news media activity, social media (especially Twitter) was the primary driver of the Bela campaign, with only a limited number of online outlets publishing stories about the campaign.

Having an existing Bela user base proved extremely helpful, as they helped share news of the campaign. Some existing Bela users backed the campaign to acquire additional hardware. To maintain attention and interest, we posted five updates during the campaign, including new demo videos, examples of other people's work using Bela, and after the campaign reached its funding target, details of two "stretch goal" accessories that would be made if the campaign funding hit certain targets.

The Kickstarter campaign raised over 1000% of its goal; nearly 300% was raised in the first 24 hours alone. Both of the advertised stretch goals were reached.

### Community Profile

A total of 530 people backed the campaign, of whom 479 selected one of the Bela kit rewards (the remainder were small pledges or backers dropped for credit card failures). As part of the Kickstarter survey to capture shipping address details, we also asked about the projects each person intended to create and the programming languages they intended to use<sup>8</sup>.

Unsurprisingly, musical applications predominate: 300 of 462 responses where the intended application was known mentioned music or audio applications, with musical instruments and synthesisers being the most common. Art installations were another frequently mentioned application, which may partly overlap with music. Other responses were more generic, e.g. "DSP" (13 responses), "teaching" (10), "prototyping" (7), "fun" (8). Some applications seem inconsistent with the capabilities of Bela, e.g. "visualisation" (3) (Bela does not include graphics capability). These results suggest that the Bela community strongly overlaps with existing communities building DMI and other audio systems, and may suggest that people are interested in Bela for similar reasons.

The most commonly mentioned programming language was also the most low-level: C/C++ (277 responses out of 752; some respondents indicated more than one language). The graphical Pd language was second at 219 responses, with significant interest in community-contributed FAUST (70) and Python (23) support. SuperCollider (SC), a popular music programming language, was mentioned by 25 respondents. In fact, SC support was only announced 4 months after the campaign finished, though Twitter activity during the campaign indicated that early experiments were underway; 14 respondents indicated they would use Max/MSP, which is impossible to run on Bela; 22 other responses also named incompatible languages. The preponderance of users naming C and C++ suggests that the Kickstarter backers represent a technically proficient subset of the digital music community. The result

<sup>8</sup>Kickstarter prohibits collecting demographic data in backer surveys, so age and gender breakdowns are unavailable. Informal analysis of the names suggests that a large majority of backers were male.

stands in contrast to the workshops we organise, where Pd is the most popular language for developing on Bela.

In October 2016, we made Bela available for general online ordering. The community has continued to grow, reaching around 700 people as of January 2017.

## COMMUNITY ACTIVITIES

### Online Resources and Forum

A set of wiki pages on Github serves as a curated focal point for instructions and documentation. Additionally, an online forum was made available to the community at the end of June 2016, alongside the shipment of the first Bela boards to Kickstarter backers. At this writing there are 167 registered users, 94 of which have posted at least once, and 35 of which have been active in the last 30 days. There are no data available on the number of non-registered users who view the forum.

Most of the 184 discussions were started by users looking for help (164); notable exceptions are those started by members announcing features or workflows they developed and wanted to share with the community (8) and those started by non-users asking for more details about the platform (6). Answers to inquiries were provided mainly by members of the Bela team, and only occasionally it was a community member who answered another member's question.

The early picture of forum usage shows that user-user interaction is still limited, with the team members still playing a major role. This can be expected from a young community where even those willing to share their findings may first have to ask questions on the subject.

### Porting

Although Bela runs a Linux OS, existing audio software that runs on Linux requires code changes to take advantage of Bela's low-latency audio environment. This is usually feasible, but it can require significant developer time.

When Bela was released, it supported C++ and Pd. Beginning in late 2015 and continuing after the Kickstarter launch, we were contacted by several developers offering to port other audio programming languages to Bela. These offers were both welcome and unexpected according to our original release plan. We discuss the implications of porting established software later in this paper as a case of *pluggable communities*.

The first language ported was FAUST<sup>9</sup>, a functional programming language for DSP which has acquired a large user base in the 10 years since its introduction. The FAUST developers contacted us one day after the Kickstarter launch to express interest in adding Bela support, inspired by experiments by a Bela early adopter [OL] outside the university. The developer of pyo<sup>10</sup>, a Python module for DSP, also contacted us during the Kickstarter campaign about porting. In both cases, we supplied hardware and expertise on the core code, and the developers adapted their platforms to run on Bela.

SC support was offered by a QMUL researcher in digital audio [DS], who is also a longtime SC contributor. He had

<sup>9</sup><http://faust.grame.fr>

<sup>10</sup><http://ajaxsoundstudio.com/software/pyo/>

earlier used Bela in one of our workshops and developed a proof-of-concept SC/Bela driver in early March 2016 to use for a personal project. Support was later extended by two participants in the workshop discussed in the next section.

Pd support was originally provided through the Heavy Audio Tools<sup>11</sup>, an online, free but closed-source compiler which can turn most Pd patches into optimised C code, providing high performance on low-powered platforms. Heavy support for Bela was developed by a member of the Bela team to use in his own research. After launching the Kickstarter, we added broader Pd support using libpd<sup>12</sup>. Heavy achieves more efficient performance, but libpd has the advantages of full support for the language and easier offline development. It is also fully open-source, which was particularly important to some members of the Pd community who felt that Heavy was incompatible with Bela’s claim to be open-source.

Another early supporter of Bela attempted to port the code generated by the gen~ object in Max<sup>13</sup>, a commercial flow-based programming language similar in concept to Pd. gen~ allows the design of high-performance DSP objects from a graphical patcher, generating C++ code. The generated code is licensed with an open license and could thus be embedded into the Bela environment, while the rest of Max could not, as its source code is not available. Later on, the developers of Max provided official support for running gen~ code on Bela.

## STEIM WORKSHOP

The Kickstarter survey provided a broad but shallow overview of why people chose to use Bela, and we had collected other feedback via emails, forum posts and personal conversations. After the campaign, we sought a way to provide a deeper (if narrower) study of the dynamics of the emerging community.

To gain this insight, in August 2016 we organised a workshop in Amsterdam at STEIM (the Studio for Electro-Instrumental Music), a renowned centre for experimental music. STEIM was chosen for its influence in the DMI community and the fact that several members had expressed interest in Bela (the authors have no affiliation with STEIM). The workshop lasted 3 days, with informal social gatherings following each day. It included eight experienced DMI creators as participants, with two Bela creators [authors GM, AM] acting as technical facilitators and two other researchers [FM, AC] collecting documentary evidence. The workshop goal was for each DMI creator to port their existing instrument design to Bela.

The advantages of the design workshop as a research practice have been discussed in literature in HCI and CSCW [37, 38] as well as in PD where this practice is suggested to be used as a tool to explore “design in use” situations [5]. In our case, we wanted to examine the motivations, concerns, practices, and priorities of the participants with whom we engaged. Questions included what technical and user experience features of Bela led them to engage, to what extent they were able to port existing designs to Bela, and the ways that participants interacted with the facilitators and with one another. The latter was

intended to examine patterns of co-working practice which would be difficult to investigate in the wild.

## Participants

Participation was limited to eight people so that the two facilitators could provide intensive, individualised technical support throughout the three days. Five participants were invited by STEIM, and the other three were selected through an open call. The selection process prioritised people who already had a functioning prototype of their instrument already running on another platform.

The final list of 8 participants (2 female: P1, P4) is shown in Table 1. The two technical facilitators introduced the platform, helped participants solve issues and suggested possible design solutions; the two other researchers collected observations and conducted interviews with participants.

**Table 1. Background, instrument, and programming language of the workshop participants.**

<i>Background</i>	<i>Instrument</i>	<i>P.L.</i>
<b>P1</b> Composer, performer	Pitched percussion DMI	SC
<b>P2</b> Composer, performer	Capacitive ceramic DMI	Pd
<b>P3</b> Composer, performer	Wearable DMI	SC
<b>P4</b> Snd artist, HW/SW dev	DMI w/pickups, effects	SC
<b>P5</b> Sound artist, designer	Augmented accordion	Pd
<b>P6</b> Musician, sound artist	No-input mixer	SC
<b>P7</b> Researcher, sound artist	Augmented accordion	SC
<b>P8</b> Musician, programmer	Chaotic oscillators	C++

## Activities

The morning of the first day, participants each presented their instrument (pre-Bela) to the group, including a description and short demo performance if the instrument was in a working state. They also outlined their plans for the workshop. Next, the facilitators introduced Bela in a presentation and supplied each participant with a Bela kit which they could keep at the end of the workshop. A hands-on tutorial in the afternoon got participants acquainted with the hardware and programming tools they would need for the rest of the workshop. The participants were aware of the key characteristics of the platform, although only two (P4, P8) had used it before.

After the tutorial, in the late afternoon of day 1, the activity shifted to instrument-making, which continued for the remainder of the three days. At the end of day 3, participants were invited to give a final demonstration or performance of their newly-ported instrument, and to reflect on their experiences with Bela with respect to their original expectations.

Seven participants worked on new instruments based on Bela; the new versions of six of the instruments are shown in Figure 3. P7 instead focused on creating core technical capabilities in SC and wrote tutorial materials which he later published online. P1, P4, and P6 stuck closest to the theme of the workshop by doing a straightforward port of a complete instrument to Bela, though P6’s port resulted in what he considered to be a new instrument. P2 and P3 used the workshop as an opportunity to rethinking their instrument designs in light of the new possibilities offered by Bela. P5 revived an old design

<sup>11</sup><https://enzienaudio.com/>

<sup>12</sup><https://github.com/libpd/libpd/>

<sup>13</sup><https://cycling74.com/>

which had been abandoned for years; P8 turned his existing software synthesis routines into a hardware instrument.

### Data Collection

Individual interviews were conducted throughout the making process and following the last presentation. Participants were invited to reflect on their motivations for using Bela and for joining the workshop; the communities they belonged to and how they contribute to them; and the potential role of Bela in their artistic practice. When not conducting interviews, the two researchers wrote observations on the workshop, focusing on the interaction among participants and with the facilitators. The performances and demos were video recorded.

### WORKSHOP DISCUSSIONS

Based on dynamics observed during the workshop and quotes collected from interviews, we reflect on the aspects that motivated participants' interest in Bela and in the workshop.

#### Motivations to Participate

##### *Team credibility*

*"People behind this project are very open to other people's opinions. Here there are efforts to open the platform to this idea of making things." ... "I don't know Andrew personally. But what I do know is that everything that he touches is substantial. If he is doing this thing it means it is good."* [P7]

These quotes from a workshop participant suggest that team credibility is as important as technological innovation to convince makers to take up a new platform. P7 explained that even before personally meeting the team members, their social presence and actions demonstrated (i) to have an indisputable expertise, (ii) to be very open to other people's opinions, (iii) "to firmly want to make this happen." P7 would not have invested his time supporting the formation of this community if one of these three aspects was lacking.

##### *Giving back to the community*

The contribution to the workshop by P4 and P7 offer interesting insights into support offered by community members.

*"I've been using SuperCollider for 12 years. Very actively. I also used this open source software from other people that invested a lot of time on it. I am just giving back to the community."* [P4]

P4 spent most of the time improving the SC port. This was the first field test of their integration, so she could observe through her work and the work of others what features were most needed and implement them straight away. This allowed her and other participants to immediately benefit while at the same time contributing back to the broader online community. P7 spent the first day trying to figure out how to connect a rotary encoder to Bela. After achieving the solution, he almost exclusively dedicated the rest of the workshop to refine the idea. He then created a tutorial, including code and diagrams, for the community.

##### *Collocated support*

*"My interest in being here is that there are people that I can ask. Programming all the things and making it together is something that I can do by myself. I have to be here for how do I get this thing to work in a reasonable way."* [P7]

The collocated work experience offered participants the possibility to make requests to the team, who in turn could learn what the wishes of the community were.

Some of the participants required features that were not implemented in the Bela. For instance, P2 was using a capacitive sensing device which previously ran on an Arduino. The particular characteristics required by this device were not supported by the Bela API. The facilitators adapted the existing code and made it run successfully on a microcontroller that was not easily accessible to the user at that moment.

The facilitators took the opportunity to integrate this functionality back in the Bela API to provide easy access to it for future users. Similarly, P5 was using an ultrasonic ranging device, which came with some example Arduino code. The facilitators helped port it to Bela and later made it available to the Bela community in the form of example code.

In other cases, collocated support came from the participants themselves. P1 received assistance on hardware and SuperCollider (SC) from P4 and P7; P4, who leads Bela SC porting efforts, even added new features to the SC code for P1. P4 also gave occasional SC assistance to P3; P6, a STEIM member, received hardware assistance from another member.

#### Technological Innovation

*"For the past N years there is a sort of revolution ingoing. In particular the BeagleBone, and Raspberry Pi. But there has always been issues with those."* [P7]

The existence of technological limitations in existing devices was the main motivation for participants to use Bela, in particular because it allows to embed all the technology in a self-contained device, with a small latency between audio input and output and a wide range of high-quality connections, in a reliable all-in-one design.

##### *Self-containedness*

*"I build a lot of sound installations and I want them to be stand-alone. They can't malfunction or shut down at one point. And I just need connect to the stereo plug."* [P2]

One of the features of Bela that most attracted interest was the possibility to have a fully functional self-contained version of their instrument that is easy to set up (P1, P2, P3, P4, P7, P8).

Bela is not the first nor the most powerful embedded device suitable for building stand-alone instruments. Other embedded solutions exist, some of them even smaller in size. However, Bela integrates all the hardware in a single device. It is more powerful than the cheaper and smaller Teensy<sup>14</sup>, but has more connectivity options than the more powerful Raspberry Pi 3, also providing more robust audio performance than the latter.

P2 reflected on this point, stating that stability is indeed an issue with other systems. For instance, he considered Raspberry Pi not very stable due to the dependency on external peripherals (e.g. Arduino).

*"Sometimes you don't want to update your laptop because you don't want to screw things".* [P2]

<sup>14</sup><https://www.pjrc.com/teensy/>



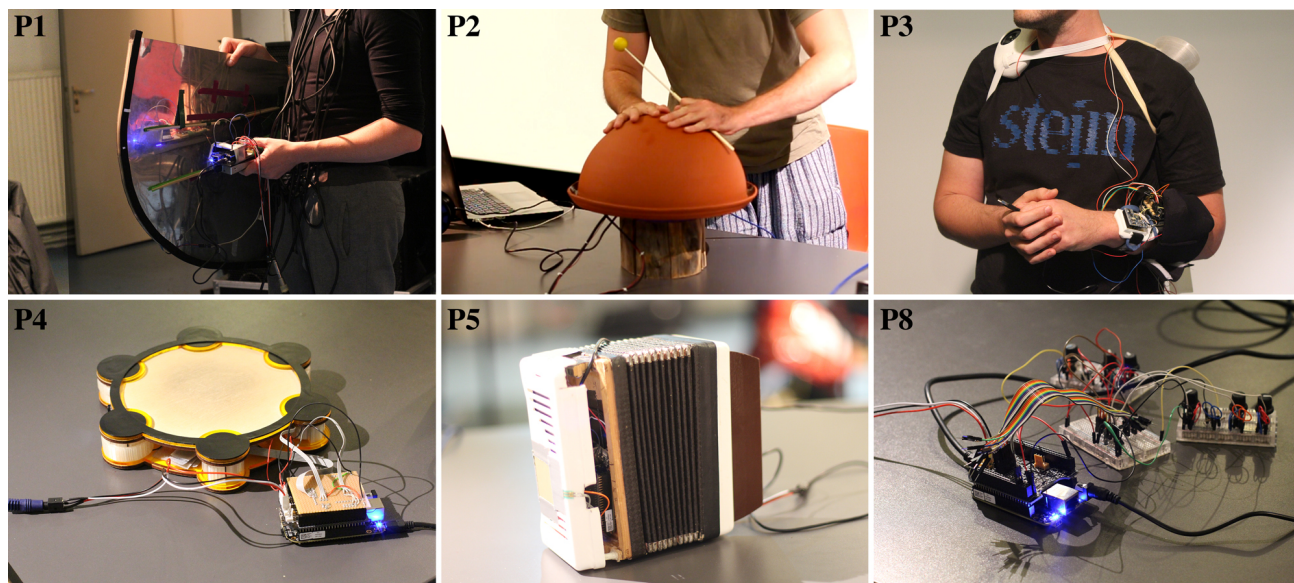


Figure 3. Six instruments that were ported to Bela during the three-day workshop at STEIM.

Traditionally, the software part of most of DMIs sits on musicians' personal laptops. When updating the laptop to new versions of operating systems, audio software, or programming languages, incidental issues often occur. Several participants noted that dedicating a piece of hardware specifically to an instrument frees them from such software upgrade issues.

#### Low latency

*"The low latency is very important for my instrument. It is a percussive instrument; I don't want to experience any latency."* [P6]

The original development of Bela was driven by the need to guarantee a low-latency feedback loops between hardware and software. Five participants (P2, P3, P4, P5, P7) attributed great importance to the low latency, which is not available in other embedded computer audio environments.

#### Improved User Experience

In addition to technical improvements, the results of the interviews suggest that Bela uptake can be explained in terms of the improved user experience for the developer.

#### Support for different programming languages

*"It is super nice that I can run SuperCollider on Bela, because it's what I am used to"* [P6]

One crucial feature that popularised Bela within four workshop participants (P3, P4, P5, P7) is the support for multiple programming languages. As opposed to other commercial alternatives, Bela supports programming with some of the most typical audio programming languages. This lets a maker seamlessly port their existing patches and compositions to Bela without learning a new language.

#### Better coding experience

*"The IDE is amazing. In terms of design process is like 100 times faster than the Raspberry Pi. I made a whole new*

*instrument in one day. Before I had to connect filters, upload new patches, rewrite the start script, the shut-down script, reboot the board, see if all works. The design process should be a playable thing if you want to be creative."* [P2]

The quality of the development environment improved the coding experience, including its rapid setup without special configuration. In particular, participants acknowledged that the browser-based IDE is a key difference compared to other embedded devices.

#### IMPLICATIONS AND RECOMMENDATIONS

Based on our experience with Bela, here we discuss characteristics we believe are important to allow for a community of makers to exist and develop around a platform. We suggest that the growth of open-source communities is not always a gradual expansion around a single tool; sometimes they grow in discrete leaps enabled by connections between existing popular tools, a process we describe as *pluggable communities*.

#### Design Factors Promoting Community Formation

##### "Make an instrument, not a platform"

In an influential paper [12], Perry Cook recommended that DMI designers "make a piece, not an instrument or controller." An analogous recommendation could be made for creators of open hardware platforms. Designing for one specific and immediate application can provide focus; designing general capabilities based on hypothetical future use cases risks missing out on essential but non-obvious features or failing to provide a unique selling point.

At every step, Bela's evolution was driven by an immediate need: the D-Box; a DSP class; specific research projects. Community contributions, too, were driven by immediate applications. Porting FAUST, pyo, and SC allowed those developers to use their existing work on Bela. The dual roles of SC-porters

P4 and P7 as instrument designers and supporters of other participants show the fluid relationship between developing one's own work and making tools for the community.

#### *Low barrier to entry, with openness to existing tools*

Our results suggest that a key component for the success of an open hardware platform is a low barrier to entry. Barriers to entry take two forms: setup time and learning curve. Based on feedback from the STEIM workshop and from Kickstarter backers, users find value in the simplicity of getting started, where only a browser is needed and the first project can run after a single click. Even seasoned engineers may not have the time to install elaborate compiler toolchains or navigate a complex developer UI.

The idea of “learning curve” is typical of HCI [40] and DMIs [6, 36]. Becoming familiar with new languages or environments takes time. Whether or not Pd, SC, or FAUST are ideal tools for creating musical instruments is less important than the fact that large groups of people already know how to use them. The time spent by the Bela creators and early adopters to port these programming languages allowed designers to seamlessly port their existing instruments to Bela.

The fact that Bela is Linux-based makes porting open-source software easier. But running an open-source operating system is not a sufficient condition: most open-source software can also be built on the Raspberry Pi platform, but the lack of a ready-made toolchain and a single convenient IDE makes the process more cumbersome. The Raspberry Pi also cannot match Bela's audio latency or connectivity, which users saw as important both on Kickstarter and at the STEIM workshop.

A lesson for open-hardware makers is that open platforms should allow future extensions that were not originally planned. These extensions might take the form of complete existing software tools which have their own established communities.

#### *Signature features: same category, new capabilities*

LilyPad Arduino became popular because it enabled makers to build artefacts that were difficult to make with prior tools [8]. New hardware platforms succeed where they contribute to expanding the creative possibilities of the maker [9]. This may seem obvious, but less is known about what kinds of new features will attract a user base.

We suggest that a new platform should have *signature features*: simple, clear capabilities that go beyond what was easily achievable before. For LilyPad it was the physical layout of the board, making it suitable for wearable computing. For Bela, we found that *1ms latency* and *self-containedness* were the most important signature features which initially attracted users. Representative Kickstarter survey comments included: “*this is stamped in my brain: 1 MILLISECOND*” and “*I've been waiting for a (programmable) platform with low latency for a long time*”. At the STEIM workshop, in addition to latency, 6 of 8 participants cited self-containedness as a reason for participating. Feedback from early adopters has been similar.

At STEIM and amongst other users, many people have described plans to port their existing work to Bela. Since the existing work is presumably already functional, it is worth

reflecting on why they would choose to reimplement it at all. Beyond the features above, reasons given by workshop participants included the value of being free from the computer, possibilities for future expansion (P3) and improved sensing performance (P2); a broader theme was that the instruments would respond better in performance once on Bela, with a corresponding creative benefit.

This result indicates that new creative possibilities do not only result from new categories of technological tool (e.g. Kinect, Leap Motion), but also from major improvements to existing technologies.

#### *Open source: necessary but not sufficient?*

During the Kickstarter, the somewhat hostile reaction of parts of the Pd community to Heavy (a free but closed tool for compiling Pd patches to C) demonstrated that having a purely open-source platform was seen as an important value, at least for certain users. However, our experience suggests that open-source materials are not themselves sufficient for a platform to attract a community. To date, the most significant contributions from Bela users have all been of the form of porting established tools rather than editing the core Bela code or building custom environments from scratch on the Bela C/C++ API.

If Bela were open source but not easily connectible to familiar tools, it is unclear whether developers would take the time to understand the internal operation of its code well enough to contribute to its development. Indeed, GitHub and similar sites have millions of open-source projects, of which few receive contributions from beyond the core developers.

### **Activities Promoting Community Formation**

#### *Crowdfunding success as cause (and effect) of community*

Related work shows that crowdfunding helps establish a connection with a community [15] and build a user base [14]. Our experience supports these findings, but shows that the relationship between crowdfunding and community is bidirectional.

For Bela, the existing base of early adopters was important to raising awareness of the platform and campaign. The credibility and social connections of the design team and early adopters were also important (e.g. P7's comments at the workshop). Studies analysing why supporters back campaigns show that they are motivated to support people they know and trust [14], and this trust can be strengthened by face-to-face contact in situated co-working activities [18].

#### *Support to and from the community*

The online forum is the primary hub for a geographically distributed community to connect. At the moment, many but not all of the discussions involve community members receiving support from the Bela creators. The public nature of the forum, though, means that others can learn from these interactions when they encounter similar questions; for example, in the workshop, P7 said that he frequently read the forum but did not post. Similarly, in the hands-on tutorials and early workshops, most support flowed from the Bela creators to the community.

The STEIM workshop offers a case study in an emerging dynamic, where community members become both users and supporters. Particularly where new programming languages

are ported to Bela, the people porting them often assume a supporting role for others in the community. P4, P8, and an early adopter [DS] mentioned giving back to the community as a motivation for their activities. Other motivations could include growing the user base for specialised languages, or making a contribution back to an established software community by porting to Bela.

Given the spontaneous involvement of Bela users in the life of the community it would be inaccurate to consider them simply “users” that happen to use the same platform. They offer support to each other (e.g. porting programming languages, exchanging ideas on the online forum), they introduce the platform to colleagues, and they share similar interests, passions and concerns (e.g. limitations of existing platforms) that are typical traits of a community of practice [48].

This spontaneous involvement is an important step toward community sustainability. The community is also currently supported by expert developers readily available to address user issues. Long-term sustainability of open-source communities depends on reducing reliance on the original developers, as many music programming languages have done (Pd, SC, Csound). For open hardware, sustainability may also imply involvement of community members in the production and distribution process. Given the early stage of Bela, we cannot yet assess whether this will be the case.

### Pluggable Communities

A central theme in the preceding discussion is the relationship between Bela as an open hardware platform and the variety of music programming languages which have been ported to it. Though Bela has its own C/C++ API, a large portion of the Bela community has come in by way of Pd, SC, and other languages which were later additions. Likewise, during the Kickstarter campaign, social media forums related to existing tools (especially Pd) were a significant source of attention.

These results suggest that the Bela community has not emerged solely from a collection of isolated individuals drawn to Bela’s core features. Rather, empowering users to map their knowledge onto a new platform enables a connection between two previously disparate communities. This connection can be facilitated by so-called *social experts*, i.e. mentors that enable a transfer of knowledge that can otherwise be quite difficult to perform [39]. For the case study of Bela and SC, P4 filled this role; she had already worked with early adopter [DS] and is known in the SC community. She also previously worked at STEIM and knew many of the participants.

Under this notion of *pluggable communities*, both open hardware and open source benefit from this connection, growing their user base and expanding their range of applications. Connecting existing tools also represents a design process which understands the needs and concerns of people using them: porting SC to Bela rather than building a new language recognises that SC has evolved over many years to suit the needs of its community, and that significant expertise has developed.

Open-source software may encourage pluggable communities insofar as it facilitates porting and community maintenance.

However, the commercial Max gen~ environment has a significant user community which, thanks to official support, can now connect to Bela. Whether future sustainability depends on open-source status remains to be seen.

### CONCLUSION

This paper has examined the process by which a maker community forms around a new open hardware platform, drawing on a longitudinal study of Bela and a follow-up 3-day workshop organised to better understand the community dynamics. We find that technical merit, user experience and release of open-source materials are important, but that they are not sufficient conditions for community formation.

Instead, social factors play a significant role, including listening to community needs and remaining accountable to the community in all parts of design, development and delivery. For Bela, community needs were initially reflected through design stages that focused on specific immediate applications (the D-Box, teaching DSP, research projects) rather than on hypothetical future uses. Later, meeting community needs became tightly intertwined with support for established music programming languages, which themselves reflect the needs of their user communities.

We describe this mutually beneficial process of connecting existing open-source tools as *pluggable communities*. We believe this concept extends beyond this specific case study and may be particularly common in open-source software. For example, the GNU/Linux ecosystem is a web of interconnected open-source tools, many of which predate the Linux kernel.

Other results presented in this paper are generalisable as pragmatic recommendations to HCI researchers and practitioners. Completely new categories of tool are not necessarily required for users to find a creative benefit, and in fact a connection to established working practices is helpful to attracting new users. Easily identifiable *signature features* may compel users to try a new platform, and maintaining low barriers to entry and a suitable support system may be necessary conditions for them to stay with it. Other findings, including the recommendation to make a specific instrument rather than a general platform, may apply mainly within the digital music and art domains.

The term “maker community” can relate to either a general worldwide movement interested in DIY practices or to domain-specific communities of interest [1, 7, 26]. An open question is whether the Bela community yet represents a definable maker community of its own, or whether it remains merely a subset of the broader digital musical instrument (DMI) community. Time will tell whether Bela acquires the characteristics of an internally cohesive community, and how that community will ultimately influence its future development.

### ACKNOWLEDGEMENTS

Thanks to the workshop participants and the Bela community for supporting the project and providing insights, and to EPSRC for funding this research under grants EP/N005112/1 (Design for Virtuosity) and EP/L019981/1 (Fusing Semantic and Audio Technologies).

## REFERENCES

1. Morgan G Ames, Jeffrey Bardzell, Shaowen Bardzell, Silvia Lindtner, David A Mellis, and Daniela K Rosner. 2014. Making cultures: empowerment, participation, and democracy-or not?. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1087–1092.
2. Chris Anderson. 2010. The new industrial revolution. *Wired Magazine* 18, 2 (2010).
3. Jeffrey Bardzell, Shaowen Bardzell, Lilly Irani, Silvia Lindtner, Kaiton Williams, and John Zimmerman. 2016. Boundary Troubles: Here, There, Design, Make, Research. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1051–1056.
4. Edgar Berdahl and Wendy Ju. 2011. Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform.. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 173–178.
5. Erling Björgvinsson, Pelle Ehn, and Per-Anders Hillgren. 2012. Design things and design thinking: Contemporary participatory design challenges. *Design Issues* 28, 3 (2012), 101–116.
6. Tina Blaine and Sidney Fels. 2003. Collaborative musical experiences for novices. *Journal of New Music Research* 32, 4 (2003), 411–428.
7. Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 423–432.
8. Leah Buechley and Benjamin Mako Hill. 2010. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*. ACM, 199–207.
9. Leah Buechley and Hannah Perner-Wilson. 2012. Crafting technology: Reimagining the processes, materials, and cultures of electronics. *ACM Transactions on Computer-Human Interaction (TOCHI)* 19, 3 (2012), 21.
10. Leah Buechley, Daniela K Rosner, Eric Paulos, and Amanda Williams. 2009. DIY for CHI: methods, communities, and values of reuse and customization. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4823–4826.
11. Joel Chan, Steven Dang, and Steven P Dow. 2016. Improving crowd innovation with expert facilitation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 1223–1235.
12. Perry Cook. 2001. Principles for designing computer music controllers. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
13. Sean Ferguson and Marcelo M Wanderley. 2010. The McGill Digital Orchestra: An interdisciplinary project on digital musical instruments. *Journal of Interdisciplinary Music Studies* 4, 2 (2010), 17–35.
14. Elizabeth M Gerber and Julie Hui. 2013. Crowdfunding: Motivations and deterrents for participation. *ACM Transactions on Computer-Human Interaction (TOCHI)* 20, 6 (2013), 34.
15. Elizabeth M Gerber, Julie S Hui, and Pei-Yi Kuo. 2012. Crowdfunding: Why people are motivated to post and fund projects on crowdfunding platforms. In *Proceedings of the International Workshop on Design, Influence, and Social Technologies: Techniques, Impacts and Ethics*.
16. Reed Ghazala. 2005. *Circuit Bending: Build Your Own Alien Instruments*. Wiley.
17. Rishab A Ghosh, Ruediger Glott, Bernhard Krieger, and Gregorio Robles. 2002. Free/libre and open source software: Survey and study. Technical report, University of Maastricht. (2002).
18. Pamela J Hinds and Catherine Durnell Cramton. 2013. Situated coworker familiarity: How site visits transform relationships among distributed workers. *Organization Science* 25, 3 (2013), 794–814.
19. Alexander Refsum Jensenius and Michael J. Lyons (Eds.). 2017. *A NIME Reader—Fifteen years of New Interfaces for Musical Expression*. Springer, Berlin.
20. Sergi Jordà. 2004. Digital instruments and players: part I—efficiency and apprenticeship. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 59–63.
21. Christopher M Kelty. 2008. *Two bits: The cultural significance of free software*. Duke University Press.
22. Robert E Kraut and Paul Resnick. 2011. Encouraging contribution to online communities. In *Building successful online communities: Evidence-based social design*. MIT Press, 21–76.
23. Stacey Kuznetsov and Eric Paulos. 2010. Rise of the expert amateur: DIY projects, communities, and cultures. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM, 295–304.
24. Stacey Kuznetsov, Alex S Taylor, Tim Regan, Nicolas Villar, and Eric Paulos. 2012. At the seams: DIYbio and opportunities for HCI. In *Proceedings of the Designing Interactive Systems Conference*. ACM, 258–267.
25. Steven Levy. 2001. *Hackers: Heroes of the computer revolution*. Vol. 4. Penguin Books New York.
26. Silvia Lindtner, Shaowen Bardzell, and Jeffrey Bardzell. 2016. Reconstituting the utopian vision of making: HCI after technosolutionism. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1390–1402.



27. Silvia Lindtner, Garnet D Hertz, and Paul Dourish. 2014. Emerging sites of HCI innovation: hackerspaces, hardware startups & incubators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 439–448.
28. Adnan Marquez-Borbon and Paul Stapleton. 2015. Fourteen Years of NIME: The Value and Meaning of 'Community' in Interactive Music Research. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
29. Andrew McPherson and Victor Zappi. 2015a. An environment for submillisecond-latency audio and sensor processing on BeagleBone Black. In *Proc. AES 138th Conv.*
30. Andrew McPherson and Victor Zappi. 2015b. Exposing the Scaffolding of Digital Instruments with Hardware-Software Feedback Loops. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
31. Andrew P. McPherson, Alan Chamberlain, Adrian Hazzard, Sean McGrath, and Steve Benford. 2016. Designing for Exploratory Play with a Hackable Digital Musical Instrument. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. New York, NY, USA, 1233–1245.
32. Andrew P McPherson and Youngmoo E Kim. 2012. The problem of the second performer: Building a community around an augmented piano. *Computer Music Journal* 36, 4 (2012), 10–27.
33. David A Mellis and Leah Buechley. 2014. Do-it-yourself cellphones: an investigation into the possibilities and limits of high-tech diy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1723–1732.
34. Iqbal Mohamed and Prabal Dutta. 2015. The age of DIY and dawn of the maker movement. *GetMobile: Mobile Computing and Communications* 18, 4 (2015), 41–43.
35. Giulio Moro, S. M. Astrid Bin, Robert H. Jack, Christian Heinrichs, and Andrew P. McPherson. 2016. Making high-performance embedded instruments with Bela and Pure Data. In *Proceedings of the International Conference on Live Interfaces*.
36. Fabio Morreale, Antonella De Angeli, and Sile O'Modhrain. 2014. Musical Interface Design: An Experience-oriented Framework. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 467–472.
37. Paul Rabinow, George E Marcus, James D Faubion, and Tobias Rees. 2008. *Designs for an Anthropology of the Contemporary*. Duke University Press.
38. Daniela K Rosner, Saba Kawas, Wenqi Li, Nicole Tilly, and Yi-Chen Sung. 2016. Out of Time, Out of Place: Reflections on Design Workshops as a Research Method. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 1131–1141.
39. Richard Sennett. 2008. *The Craftsman*. Yale University Press.
40. Helen Sharp. 2003. *Interaction Design*. John Wiley & Sons.
41. Susan Currie Sivek. 2011. "We Need a Showing of All Hands": Technological Utopianism in MAKE Magazine. *Journal of Communication Inquiry* (2011).
42. Deborah Sole and Amy Edmondson. 2002. Situated knowledge and learning in dispersed teams. *British journal of management* 13, S2 (2002), S17–S34.
43. Joshua G Tanenbaum, Amanda M Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2603–2612.
44. Nick Taylor, Ursula Hurley, and Philip Connolly. 2016. Making community: the wider role of makerspaces in public life. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1415–1425.
45. Maurizio Teli, Silvia Bordin, María Menéndez Blanco, Giusi Orabona, and Antonella De Angeli. 2015. Public design of digital commons in urban places: a case study. *International Journal of Human-Computer Studies* 81 (2015), 17–30.
46. Austin L Toombs, Shaowen Bardzell, and Jeffrey Bardzell. 2015. The proper care and feeding of hackerspaces: care ethics and cultures of making. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 629–638.
47. Erik H Trainer, Arun Kalyanasundaram, Chalalai Chaihirunkarn, and James D Herbsleb. 2016. How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 1118–1130.
48. Etienne Wenger. 1998. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press.
49. Victor Zappi and Andrew McPherson. 2014. Design and Use of a Hackable Digital Instrument. In *Proceedings of the International Conference on Live Interfaces*.